JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS: Vol. 122, No. 2, pp. 323–343, August 2004 (© 2004)

Multiple-Spacecraft Reconfiguration through Collision Avoidance, Bouncing, and Stalemate[1]

Y. Kim[2], M. Mesbahi[3], and F. Y. Hadaegh[4]

Communicated by M. J. Balas

Abstract. We consider constrained multiple-spacecraft reconfigurations outside a gravity well in deep space. As opposed to the single-spacecraft scenario, such reconfigurations involve collision avoidance constraints that can be embedded in a nonconvex, state-constrained optimal control problem. Due to the difficulties in solving this general class of optimal control problems, we adopt a heuristically motivated approach to multiple-spacecraft reconfigurations. Then, we proceed to prove the convergence properties of the proposed approach for reconfigurations involving an arbitrary number of spacecraft.

Key Words. Multiple-spacecraft reconfiguration, state-constrained optimal control, collision avoidance, bouncing, stalemate, heuristic algorithms.

1. Introduction

The distributed space system architecture has been identified as a novel paradigm for many of the future NASA (Earth and deep space), Air Force, Navy, and commercial satellite space missions (e.g., Refs. 1–8). In this paper, we consider a generic problem that is embedded in the

[1]The research of the first two authors was supported by National Science Foundation Grant CMS-0093456 and by a grant from Jet Propulsion Laboratory, California Institute of Technology. The research of the third author was carried out at Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California.

[2]Graduate Student, Department of Aeronautics and Astronautics, University of Washington, Seattle, Washington.

[3]Assistant Professor, Department of Aeronautics and Astronautics, University of Washington, Seattle, Washington.

[4]Senior Research Scientist, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California.

0022-3239/04/0800-0323/0

planning, guidance, and control of such multiple-spacecraft missions, namely, the multiple-spacecraft collision-free reconfiguration problem outside a gravity well in deep space. A unique feature of this problem, as opposed to a single-spacecraft reconfiguration (e.g. Ref. 9), is the presence of collision-avoidance constraints. Such constraints appear as a set of nonconvex state constraints when the desired reconfiguration strategy is formalized in terms of an optimal control problem (Refs. 10–14). For a dual-spacecraft system, this problem was solved recently via the optimal control framework in Ref. 15. However, as it was observed in Ref. 15, the solution methods that are based on necessary optimality conditions become prohibitively complex as the number of spacecraft involved in the reconfiguration increases. In view of these complications, in the present paper we consider a heuristically motivated, greedy-like approach to multiple-spacecraft reconfiguration. Notwithstanding, the heuristics is introduced only to motivate the solution methodology; as we will see in Section 3, the parameters in the appproach can be chosen to guarantee the convergence of the resulting algorithm.

The paper is organized as follows. In Section 2, we formalize, discretize, and then reformulate the multiple-spacecraft reconfiguration problem. The heuristic approach and the resulting algorithm are described, first informally and then more precisely in Section 3. The convergence properties of the heuristic approach are elaborated on first in Section 3.3 and then in Section 3.5. Simulation results demonstrate the behavior of the proposed algorithm for a representative mission scenario in Section 4.

A few words on the notation. For two sets A and B, $A\backslash B$ denotes the set of elements that are in A but not in B. The inertial translational state (position and velocity) of spacecraft i and the control force acting on spacecraft i at time instant k are denoted by

$$x_i(k) := [p_i(k), v_i(k)]^T \in \mathbb{R}^6$$

and $u_i(k)$, respectively. Moreover, we set

$$u(k) := [u_1(k), u_2(k), \ldots, u_n(k)]^T \in \mathbb{R}^{3n}$$

to represent the multiple-spacecraft control vector at time instant k. The 2-norm of the vector x is denoted by $||x||$; we use $\langle x, y\rangle$ for the inner product of two vectors x and y. For a real number x, $\lceil x \rceil$ is the least integer greater than x. Finally, $I_{n\times n}$ and $0_{n\times n}$ designate the $n \times n$ identity matrix and the zero matrix, respectively.

2. Problem Statement: Discretization and Reformulation

The problem considered henceforth is as follows: the initial inertial and desired relative translational states (positions and velocities) of the multiple-spacecraft system are given. We are interested in specifying the control forces that steer the space system to desired relative states, while keeping away a certain minimum distance between each spacecraft pair during the required maneuver.[5] We note that the desired inertial translational states for the multiple spacecraft are not specified a priori. Instead, each spacecraft is allowed to assume an arbitrary inertial terminal state that is consistent with the desired relative state specifications. Therefore, the collision-free multiple-spacecraft reconfiguration problem is that of specifying the control forces

$$u_i(t), \quad i=1,2,\ldots,n, \quad t\in[t_0,t_f],$$

subject to the dynamic constraints[6]

$$\dot{x}_i(t)=Ax_i(t)+B_iu_i(t), \tag{1}$$

the initial and final relative positions

$$x_i(t_0), x_j(t_0) \quad x_i(t_f)-x_j(t_f), \tag{2}$$

and finally, for a given $\rho_{ij}>0$, with $i,j=1,2,\ldots,n$ and $i\neq j$, the collision-avoidance constraints

$$\|C\{x_i(t)-x_j(t)\}\|\geq\rho_{ij}, \quad i\neq j, \quad t\in[t_0,t_f], \tag{3}$$

where

$$A:=\begin{bmatrix}0_{3\times3} & I_{3\times3}\\ 0_{3\times3} & 0_{3\times3}\end{bmatrix}, \quad B_i:=(1/m_i)\begin{bmatrix}0_{3\times3}\\ I_{3\times3}\end{bmatrix}, \quad C:=\begin{bmatrix}I_{3\times3} & 0_{3\times3}\end{bmatrix}.$$

The collision-avoidance constraints necessitate implicitly that

$$\|C\{x_i(t_0)-x_j(t_0)\}\|\geq\rho_{ij}, \quad \|C\{x_i(t_f)-x_j(t_f)\}\|\geq\rho_{ij},$$

for all $i,j=1,\ldots,n$ and $i\neq j$.

[5] In this paper, we do not address the fuel/time optimality properties of the corresponding control forces.

[6] This corresponds to modeling each spacecraft as a double integrator; see Ref. 15 for the justification of such a simplification.

Before we describe our heuristically motivated approach to this problem, let us employ "Euler's first-order" discretization method and then reformulate the reconfiguration problem in terms of the relative state vectors among the multiple spacecraft. Thus, for each time instant k, we denote the relative state of spacecraft i (with respect to a reference frame attached to spacecraft j) by

$$x_{i/j}(k) := x_i(k) - x_j(k) = [p_{i/j}(k), v_{i/j}(k)]^T \in \mathbb{R}^6.$$

Now, for each $k = 0, 1, \ldots, N-1$, consider the optimization problem

$$\min_{u(k)} \sum_{i=1}^{n} \sum_{j=1}^{n} \|C\{x_{i/j}(k+2) - x_{i/j}(N)\}\|^2, \tag{4}$$

subject to the dynamic constraints

$$x_{i/j}(k+1) = (I_{6\times 6} + sA)x_{i/j}(k) + B_{ij}u_{i/j}(k), \tag{5}$$

where $i, j, = 1, 2, \ldots, n$ and $i \neq j$, the parameter s is the sampling time (i.e., $Ns = t_f - t_0$), and for each k,

$$u_{i/j}(k) := u_i(k) - u_j(k) \in \mathbb{R}^3;$$

in Eqs. (4)–(5), the initial and final relative states are specified as

$$x_{i/j}(0), \;\; x_{i/j}(N), \tag{6}$$

the collision constraints as

$$\|Cx_{i/j}(k)\| \geq \rho_{ij}, \quad i \neq j, \;\; k = 0, 1, \ldots, N, \tag{7}$$

and finally

$$A = \begin{bmatrix} 0_{3\times 3} & I_{3\times 3} \\ 0_{3\times 3} & 0_{3\times 3} \end{bmatrix}, \;\; B_{ij} = \begin{bmatrix} 0_{3\times 3} & 0_{3\times 3} \\ (1/m_i)I_{3\times 3} & (-1/m_j)I_{3\times 3} \end{bmatrix},$$
$$C = \begin{bmatrix} I_{3\times 3} & 0_{3\times 3} \end{bmatrix}.$$

Referring to (4), we note that the quantity $\|C\{x_{i/j}(k+2) - x_{i/j}(N)\}\|^2$ is a natural choice to appear in the performance index, as one can only hope to influence the relative position between a pair of spacecraft through the application of the control force within a two-step lag. Moreover, pertaining to the dimension of the optimization problem (4)–(7), we note that, for a fixed index m, if the relative states $x_{m/j}$, $j = 1, \ldots, n$ and $j \neq m$,

have already been specified, then all other relative states $x_{i/j}$, $(i, j = 1, \ldots, n$ and $i \neq j)$ are automatically determined via the identities

$$x_{i/j}(k) = -x_{m/i}(k) + x_{m/j}(k), \quad k = 0, 1, \ldots, N.$$

Thus, only $n-1$ (rather than $n^2 - n$) dynamic constraints (5) need to be included in the optimization problem (4)–(7) at each time step. In fact, for some fixed index m, the multiple-spacecraft reconfiguration problem can be reformulated as the reconfiguration of $n-1$ relative states $x_{m/j}$, $j = 1, 2, \ldots, n$ and $j \neq m$, while avoiding the constrained regions specified by the inequalities (7) with $i, j = 1, \ldots, n$ and $i \neq j$.

3. Heuristic Approach

The solution method considered in this paper is inspired by the behavior of an elastic ball as it travels across a vertical column consisting of obstacles; we will refer to the resulting procedure as the bouncing ball (BB) algorithm. In this section, we first examine the issues that need to be examined in such a framework via an example. Then, we proceed to informally describe the BB algorithm.

Figure 1 serves as the starting point to develop the bouncing ball heuristics. In this figure, spacecraft i and j are in the midst of a reconfiguration while avoiding the spherical constraint set defined by ρ_{ij}. It seems natural that, at the outset of the maneuver, both spacecraft choose control

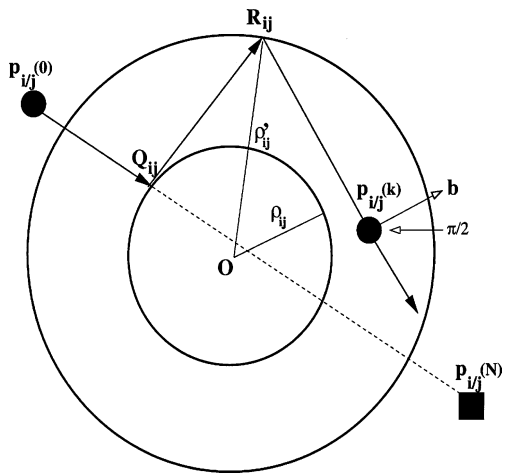

Fig. 1. Motivating the BB Algorithm.

forces that are consistent with having the relative position $p_{i/j}$ follow a straight-line trajectory (the solid line in the figure) from $p_{i/j}(0)$ to $p_{i/j}(N)$; this can be done, for example, in an optimal fuel efficient or time efficient manner. However, it can very well be that implementing such a strategy guides the relative position vector $p_{i/j}$ to violate the collision constraint (7); this situation is indicated by having

$$p_{i/j}(k) = Q_{ij}, \text{ for some time index } k,$$

in Fig. 1. In this case, the control forces on the two spacecraft are chosen such that any further violation of the collision constraints is avoided, causing the relative state vector $p_{i/j}$ to follow the solid line, instead of the dashed line. We will refer to such a sudden change in the direction of the control force as bouncing. An interesting complication that can potentially occur following a bounce is that avoiding the first constraint violation might lead to yet another relative position conflict; this issue will be considered in Section 3.1. For now, let us proceed assuming that this complication does not arise. Thus, the two spacecraft move away from the constrained region until

$$p_{i/j}(k) = R_{ij}, \text{ for some time index } k,$$

in Fig. 1. At this point, the two spacecraft are again allowed to choose control forces that are consistent with following a straight line trajectory (the solid line) to reach the desired relative position $p_{i/j}(N)$, this time starting from the relative position R_{ij}.

3.1. Bouncing Ball Algorithm. We now present a more refined, yet still informal, description of the bouncing ball (BB) algorithm for n-spacecraft reconfiguration. We include also remedies for the complication that we pointed out in Section 2.

Algorithm BB (Informal Description)

Initialization. Let $k = 0$ and $\nu = 1$.

Termination. For a prespecified, sufficiently small $\epsilon > 0$, if

$$\|x_{i/j}(k) - x_{i/j}(N)\| \leq \in, \quad i, j = 1, \ldots, n \text{ and } i \neq j,$$

terminate the algorithm.

Phase I. Moving from $p_{i/j}(k) \to p_{i/j}(N)$ or $p_{i/j}(k) \to Q_{ij}$ in Fig. 1. At the time instant k, if none of the constraints (7) is violated, find the control vector $u(k)$ that minimizes the performance index (4). If the relative position vector $p_{i/j}(k)$ is on the boundary or inside its corresponding forbidden region at some time instant k, proceed with Phase II below.

Phase II. Find the control vector $u(k)$ minimizing the performance index (4), while ensuring that all relative velocities at the time instant $k+1$ are directed away from their respective constrained regions. If the resulting relative velocity vector $v_{i/j}(k+1)$ is nonzero, apply the control $u(k)$ and proceed with Phase III below. However, if the resulting relative velocity vector $v_{i/j}(k+1)$ turns out to be equal to zero, implying that the vector $p_{i/j}$ is stationary at the point Q_{ij} in Fig. 1, proceed with Phase II – Stalemate below.

Phase II – Stalemate. The aim of this phase is to reconfigure the multiple spacecraft such that $v_{i/j}(k+1)$ can assume a nonzero value in Phase II above. For this purpose, the algorithm proceeds by finding first the νth largest magnitude relative position, say $p_{i/j'}$ at time instant k. Then, a control force is chosen ensuring that

$$v_{i/j}(k+1)=0, \quad \text{for all } j\neq j', j\neq i,$$

and $v_{i/j'}(k+1)\neq 0$ is directed away from its associated constrained region. When the magnitude of $p_{i/j'}$ reaches a prescribed value away from its associated constrained region, the algorithm proceeds with Phase II above. If the algorithm still obtains a zero value for the relative velocity vector $v_{i/j}(k+1)$, we let $\nu=\nu+1$ and invoke Phase II-Stalemate again. We note that, after at most $n-2$ calls to Phase II-Stalemate, all relative position vectors are sufficiently away from $p_{i/j}(k)$ so that $v_{i/j}(k+1)$ can eventually assume a nonzero value. For the example shown in Fig. 2, with $n=6$ and $i=6$, implementing Phase II-Stalemate results in having, for some time index k, the identities

$$p_{6/j}(k)=T_j, \quad j=1,2,\ldots,5.$$

Phase III. Moving from Q_{ij} to R_{ij} in Fig. 1. Let the control force $u(k)=0$, implying that every relative velocity remains constant, until $p_{i/j}$ assumes the value R_{ij} on the boundary of the sphere associated with the equation $\|Cx_{i/j}\|=\rho'_{ij}$ (see Fig. 1); the algorithm then proceeds with Phase I above.

In Section 3.5, we will show that the value of the parameters $\rho'_{ij}, i, j=1,\ldots,n$ and $i\neq j$, in Phase III of the BB algorithm can be chosen such that every relative position vector is guaranteed to experience a finite number of bounces during the entire reconfiguration. In fact, the required number of bounces is a function of the number of spacecraft n and the ratios $\rho'_{ij}/\rho_{ij}, i, j=1,\ldots,n$ and $i\neq j$, implicitly selected in Phase III.

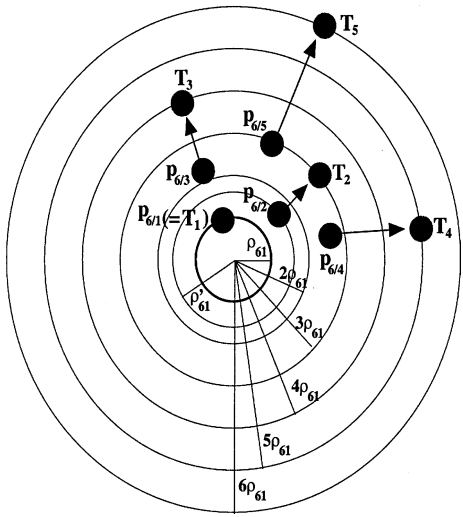

Fig. 2. Phase II - Stalemate of the BB Algorithm; the relative state $p_{6/1}$ is initially stationary on the boundary of its constrained region.

3.2. Computational Aspects of the BB Algorithm. We elaborate now on the computational aspects of the BB algorithm. We note first that, in Phases I and III, where the collision constraints (7) are not active, the control forces are obtained by solving the unconstrained quadratic program (4)–(5). However, in Phase II, where at least one of the collision constraints (7) is violated, the control forces cannot be obtained by solving a simple mathematical program, as the corresponding feasible set is nonconvex.

As an example, consider the trajectory of a particular relative state (say $x_{1/2}$, shown in Fig. 3), where the relative position vector $p_{1/2}$ is about to violate the constrained region defined by $\|Cx_{1/2}(k)\| \geq \rho_{12}$ at the time instant k; concurrently, we consider a case where the vector $p_{1/2}(k)$ is surrounded closely by the other relative states. In such a situation, it is intuitive to require that the relative velocity vector $v_{1/2}(k+1)$ satisfy

$$\langle v_{1/2}(k+1), p_{1/2}(k) \rangle \geq 0, \tag{8}$$

so that $p_{1/2}$ is not led to a further violation of the corresponding constraint; see the dashed area in Fig. 3 containing the vector $p_{1/2}(k)$. However, as we will see in Section 3.5, the constraint (8) would not guarantee the convergence of the BB Algorithm by itself; in fact, we will need an

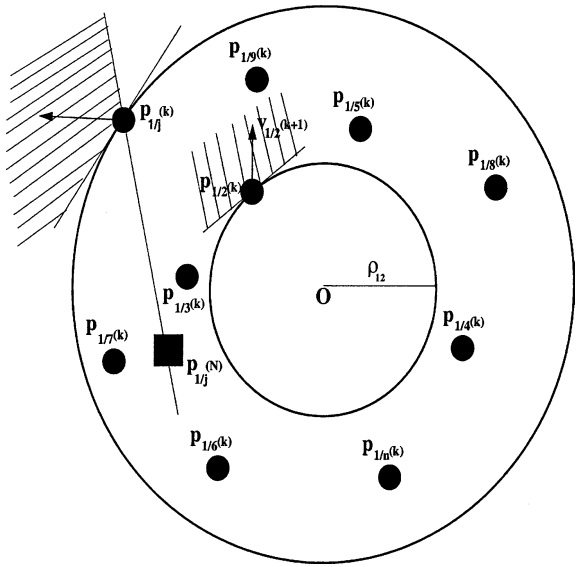

Fig. 3. Several relative states surround the relative position vector $p_{1/2}(k)$.

additional set of conditions of the form

$$\langle v_{1/2}(k+1),\, p_{1/2}(k)\rangle = 0, \qquad \langle v_{1/2}(k+1),\, p_{1/2}(N)\rangle \geq 0, \tag{9}$$

ensuring that the relative velocity vector $v_{1/2}(k+1)$ is pointed away from the constraint region and is directed toward the final position $p_{1/2}(N)$. Furthermore, we need to impose constraints on the relative velocity vectors ensuring that every other relative state $x_{i/j}$, $j = 3, \ldots, n$, avoid a violation of their respective collision constraints after the application of the computed control force $u(k)$. Thus, we are led to include the inequalities

$$\langle v_{1/j}(k+1),\, p_{1/j}(k)\rangle \geq 0, \quad j = 3, 4, \ldots, n,$$
$$\langle v_{i/j}(k+1),\, p_{i/j}(k)\rangle \geq 0, \quad i, j = 2, 3, \ldots, n, \text{ and } i \neq j.$$

We propose thereby that, when one relative state vector $x_{p/q}$ is about to violate a constraint at time instant k, the multiple-spacecraft control $u(k)$ is obtained by solving the (convex) quadratic program

$$\min_{u(k)} \sum_{i=1}^{n} \sum_{j=1}^{n} \|C\{x_{i/j}(k+2) - x_{i/j}(N)\}\|^2, \tag{10}$$

subject to the constraints

$$\left\langle D\{(I_{6\times 6}+sA)x_{p/q}(k)+B_{pq}u_{p/q}(k)\},\, p_{p/q}(N)\right\rangle \geq 0, \tag{11}$$

$$\left\langle D\{(I_{6\times 6}+sA)x_{p/q}(k)+B_{pq}u_{p/q}(k)\},\, p_{p/q}(k)\right\rangle = 0, \tag{12}$$

$$\left\langle D\{(I_{6\times 6}+sA)x_{i/j}(k)+B_{ij}u_{i/j}(k)\},\, p_{i/j}(k)\right\rangle \geq 0, \tag{13}$$

where $D := [0_{3\times 3}\ \ I_{3\times 3}]$, with $i, j \in \{1, \ldots, n\}\backslash\{p, q\}$ and $i \neq j$.

3.3. Convergence Properties: Preliminary Considerations. We proceed now to investigate further the convergence properties of the BB Algorithm described in Section 3.1. Referring to Fig. 1, we note first that, during the execution of the algorithm, all relative states $x_{i/j}$ that have reached their respective liberating points R_{ij} after a bounce, should be directed to their respective final relative states $x_{i/j}(N)$. However, one cannot rule out further relative position violations, leading to a set of possible infinitely many bounces for $x_{i/j}$. Our objective in this section is to obtain conditions under which all relative states are guaranteed to reach their final desired values within a finite number of bounces.

As an example, consider the relative state $x_{i/j}$, which has just experienced a bounce (see Fig. 4). Furthermore, suppose that another relative state (say $x_{s/t}$) violates its associated constrained set defined by $\|Cx_{s/t}(k)\| \geq \rho_{st}$, at time instant k. The quadratic programs (10)–(13), in which the vectors $x_{s/t}$ and $x_{i/j}$, satisfy (11), (12) and (13) respectively, are then solved to find the corresponding control forces. Let us presume that

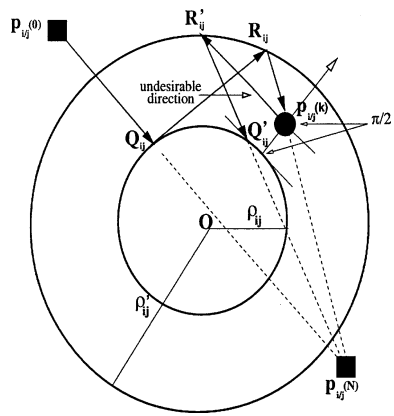

Fig. 4. Undesirable trajectory (solid lines with an arrow) for the relative position vector $p_{i/j}$.

implementing the BB Algorithm causes $p_{i/j}$ to assume initially the vector value R'_{ij} and then be directed toward its final desired value $p_{i/j}(N)$. As shown in Fig. 4, the trajectory of $p_{i/j}$ intersects the constrained region at the point Q'_{ij}, leading to yet another bounce. In fact, we note that this repeated bouncing can potentially occur infinitely many times. To rule out such an undesirable behavior, we impose a set of additional constraints on the relative velocity vector $v_{i/j}(k+1)$ in addition to (13), in order to reduce, and in fact provide an upper bound on, the total number of required bounces. In this venue, consider the inequalities

$$\langle v_{i/j}(k+1), b\rangle \geq 0, \tag{14}$$

$$\langle v_{i/j}(k+1), p_{i/j}(k)\rangle \geq 0, \tag{15}$$

where

$$b := p_{i/j}(k) + t(p_{i/j}(N) - p_{i/j}(k)), \tag{16}$$

with

$$t = -\langle p_{i/j}(k), p_{i/j}(N) - p_{i/j}(k)\rangle / \|p_{i/j}(N) - p_{i/j}(k)\|.$$

The inequality (14) represents the half-space defined by the normal vector b, as shown in Fig. 1, perpendicular to and passing through the line containing the two relative position vectors $p_{i/j}(k)$ and $p_{i/j}(N)$. The inequality (15), much in the spirit of the inequality (13), ensures that the relative state $x_{i/j}$ stays outside its constrained region. As a consequence of imposing these inequalities, the relative position vector $p_{i/j}$ lies at the intersection of the two half-spaces represented by (14) and (15), even after undergoing a bounce. As we will see in Section 3.5, this last property enables us to bound the total number of relative position bounces required during the reconfiguration.

With the vector b defined as in (16) and

$$(s,t) \in \mathcal{I} = \{(s,t) | x_{s/t} \text{ has just experienced a bounce}\}, \tag{17}$$

we are led to the following quadratic program, where it is assumed that the relative state $x_{p/q}$ is about to violate its associated constraint set at the time instant k:

$$\min_{u(k)} \sum_{i=1}^{n} \sum_{j=1}^{n} \|C\{x_{i/j}(k+2) - x_{i/j}(N)\}\|^2, \tag{18}$$

subject to the constraints

$$\left\langle D\{(I_{6\times 6}+sA)x_{p/q}(k)+B_{pq}u_{p/q}(k)\},\, p_{p/q}(N)\right\rangle \geq 0, \tag{19}$$

$$\left\langle D\{(I_{6\times 6}+sA)x_{p/q}(k)+B_{pq}u_{p/q}(k)\},\, p_{p/q}(k)\right\rangle = 0, \tag{20}$$

$$\left\langle D\{(I_{6\times 6}+sA)x_{s/t}(k)+B_{st}u_{s/t}(k)\},\, b\right\rangle \geq 0, \tag{21}$$

$$\left\langle D\{(I_{6\times 6}+sA)x_{s/t}(k)+B_{st}u_{s/t}(k)\},\, p_{s/t}(k)\right\rangle \geq 0, \tag{22}$$

$$\left\langle D\{(I_{6\times 6}+sA)x_{i/j}(k)+B_{ij}u_{i/j}(k)\},\, p_{i/j}(k)\right\rangle \geq 0. \tag{23}$$

In (18)–(23), $(s,t)\neq(p,q)$ and (i,j) in inequality (23) runs through all pairs from $i,j=1,2,\ldots,n$ with $i\neq j$ except (p,q) and the pairs already in the set $\mathcal{I}$ (17).

We note that our solution strategy for multiple-spacecraft reconfiguration involves essentially transforming the original nonconvex program (4)–(7) to a sequence of convex quadratic programs of the form (18)–(23) that can be solved efficiently via available software (e.g., Ref. 16).

3.4. Algorithm BB Revisited. We present now the formal description of the proposed BB Algorithm.

Algorithm BB (Formal Description).

Step 1. Let n be the number of spacecraft and for $i,j=1,2,\ldots,n$ and $i\neq j$, let $x_{i/j}(0)$ and $x_{i/j}(N)$ be the initial and final relative states.

Step 2. Initialize the parameters $k=0$ and $\nu=1$.[7]

Step 3. Initialize the bouncing flag BOUNCE to be inactive.

Step 4. Initialize the set of indices for which the relative states $x_{i/j}$ have just experienced a bounce ($\mathcal{I}$) to the null set.

Step 5. For $i,j=1,2,\ldots,n$ and $i\neq j$, while for a prespecified, for sufficiently small $\epsilon>0$, $\|x_{i/j}(k)-x_{i/j}(N)\|>\epsilon$:

(a) if BOUNCE = inactive and

(i) if $\|Cx_{i/j}(k)\|>\rho_{ij}$, solve (4)–(6) for the control force vector $u(k)$;

(ii) if $\|Cx_{s/t}(k)\|\leq\rho_{st}$ for some (s,t), solve (18)–(23) for the control force vector $u(k)$;

(iii) if $v_{s/t}(k+1)\neq 0$, set BOUNCE = active and add (s,t) to the set $\mathcal{I}$;

(iv) if $v_{s/t}(k+1)=0$, call the subroutine STALEMATE and then go to Step 5;

[7]Both k and ν are global parameters that are not initialized by a subroutine call.

(b) if BOUNCE = active and

(i) if $\|Cx_{s/t}(k)\| > \rho'_{st}$, solve (4)–(6) for the control force vector $u(k)$ and set BOUNCE = inactive;
(ii) if $\rho_{st} < \|Cx_{s/t}(k)\| \leq \rho'_{st}$, set $u(k) = 0$.

Step 6. For $i, j = 1, 2, \ldots, n$ and $i \neq j$, obtain $x_{i/j}(k+1)$ from equation (5) by substituting the control force vector $u(k)$ found in Step 5.
Step 7. Increment k by one and go to Step 5.

Subroutine STALEMATE

Step 1. Define the sequence $\{a_w\}_{w=1}^{n-1}$, enumerating, in an increasing order, the ratios $\|p_{s/j}\|/\rho_{st}$ for all $j \neq s$.
Step 2. Construct the sequence $\{b_w\}_{w=1}^{n-1}$ such that $b_1 := \rho'_{st}/\rho_{st}$ and, for $w \geq 2$, $b_w := \max\{b_{w-1} + 1, \lceil a_w \rceil\}$.
Step 3. If $v < n-1$:

(i) identify $j' := \{j \mid \max_{j \neq s} \|p_{s/j}\|$ and $\|p_{s/j}\| \leq b_{n-v}\rho st\}$;
(ii) find the control force vector $u(k)$ such that, for all $j \neq j'$ and $j \neq s$, $v_{s/j}(k+1) = 0$ and $v_{s/j'}(k+1)$ satisfies (21)–(22) or (23), depending on whether $x_{s/j'}$ has just experienced a bounce, while minimizing (18);
(iii) increment v by one.

Step 4. If $v = n-1$, let $j' = t$ and find the control force vector $u(k)$ such that $v_{s/j}(k+1) = 0$ for all $j \neq j'$ and $j \neq s$, and such that $v_{s/j'}(k+1)$ satisfies (19)–(20), while minimizing (18).
Step 5. Obtain $x_{i/j}(k+1)$, for $i, j = 1, 2, \ldots, n$ and $i \neq j$, from equation (5) by substituting $u(k)$ just found and increment k by one.
Step 6. While $\|p_{s/j'}\| < b_{n-v}\rho_{st}$, set $u(k) := 0$ and obtain $x_{i/j}(k+1)$, for $i, j = 1, 2, \ldots, n$ and $i \neq j$, from equation (5) and then increment k by one.

3.5. Convergence Properties Revisited. We now consider the feasibility and convegence properties of the BB Algorithm described in Section 3.4.

Proposition 3.1. The convex quadratic program (18)–(23) has a nontrivial feasible solution (e.g., not all relative velocities are zero).

Proof. Consider the following constraints that are equivalent to (19)–(23):

$$\langle v_{p/q}(k+1),\, p_{p/q}(N)\rangle \geq 0, \tag{24}$$

$$\langle v_{p/q}(k+1),\, p_{p/q}(k)\rangle = 0, \tag{25}$$

$$\langle v_{s/t}(k+1), b\rangle \geq 0, \tag{26}$$

$$\langle v_{s/t}(k+1),\, p_{s/t}(k)\rangle \geq 0, \tag{27}$$

$$\langle v_{i/j}(k+1),\, p_{i/j}(k)\rangle \geq 0, \tag{28}$$

where the vector b is defined in (16), $(s,t) \neq (p,q)$, and the indexed pair (i,j) in (28) runs through all pairs from $i, j, = 1, 2, \ldots, n$ with $i \neq j$, except (p,q) and the pairs already in $\mathcal{I}$ (17). We note that the decision variables that appear in the constraints (24)–(28) are the relative velocities, rather than the relative control forces; the required control forces can be recovered always from the variables $v_{p/q}, v_{s/t}$, and $v_{i/j}$ via equation (5).

Suppose that one of the relative states, say $x_{1/2}$, is about to violate its associated constraint, as shown in Fig. 3 [that is, $p = 1$ and $q = 2$ in (24)–(25)]. The BB Algorithm requires henceforth that every relative state at time instant $(k+1)$ is directed away from its respective constrained region. Consider another relative position, say $x_{1/j}(k)$, $j \neq 2$, that is the furthest from the associated constrained region (Fig. 3). We observe that the vector $v_{i/j'}(k+1) = 0$, for $j' = 2, \ldots, n$ and $j' \neq j$, satisfies the constraints (24)–(25). Moreover, one can always find the nonzero relative vector $v_{1/j}(k+1)$ to satisfy either (26)–(27) or (28), depending on whether $x_{1/j}$ has just experienced a bounce; this is accomplished by choosing $v_{1/j}(k+1)$ to lie in set defined by (26)–(27) if $x_{1/j}$ has just experienced a bounce (the dashed area associated with $x_{1/j}(k)$ in Fig. 3), or choosing $v_{1/j}(k+1)$ to lie in the half-space defined by (28) otherwise. Thus, the quadratic program (18)–(23) always admit a nontrivial feasible solution. □

We present now a convergence result that elevates the status of the BB Algorithm from merely being a "heuristic" approach to multiple-spacecraft reconfiguration. The proof of this result also justifies and highlights the importance of the parameters ρ_{ij} introduced in Phase III of the BB Algorithm.[8]

Theorem 3.1. Let $\eta = \rho'_{ij}/\rho_{ij}$, for all $i, j = 1, \ldots, n$ and $i \neq j$. Then the n-spacecraft reconfiguration algorithm (BB Algorithm) converges

[8] Incidentally, these parameters influence the fuel expenditure of the corresponding reconfiguration.

within at most $Kn(n-1)/2$ bounces, where K is the smallest integer satisfying

$$\sum_{k=1}^{K-1} \theta_k + \theta'_K \geq \pi/2; \tag{29}$$

in (29), the angles θ_k and θ'_k are found via the recursions

$$\theta'_k = \tan^{-1}\left[\frac{\sqrt{\eta^2-1} - \sin(\sum_{i=1}^{k-1}\theta'_i)}{1+\cos(\sum_{i=1}^{k-1}\theta'_i)}\right],$$

$$\theta_k = \sum_{i=1}^{k-1}\theta_i + 2\theta'_k.$$

Proof. The number of required bounces is obtained by calculating the angles by which a relative position vector evolves with respect to the origin of its associated constrained region. In this venue, we consider a worst-case scenario, where the initial and final relative positions are the endpoints of a diameter of the corresponding constrained region, yielding an upper bound on the number of bounces. Figure 5 shows such a situation where, for some initial and final positions $p_{i/j}(0)$ and $p_{i/j}(N)$,

$$\angle\{\overrightarrow{Op_{i/j}(0)}, \overrightarrow{Op_{i/j}(N)}\} = \pi.$$

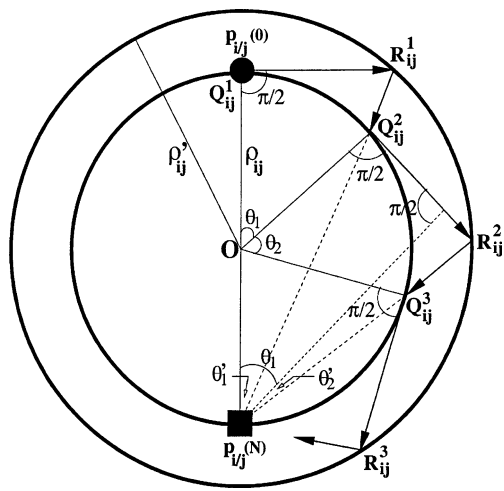

Fig. 5. Worst-case scenario for the relative position vector $p_{i/j}$ in terms of the required number of bounces.

Here, $\angle\{a, b\}$ denotes the angle between the two vectors a and b; $\overrightarrow{ab}$ denotes the vector difference $b-a$. Following the steps of the BB Algorithm, the relative position vector $p_{i/j}$ may assume subsequently the values R_{ij}^1, Q_{ij}^2, R_{ij}^2, etc. (more on this below). We proceed to determine the quantities

$$\theta_k = \angle\{\overrightarrow{OQ_{ij}^k}, \overrightarrow{OQ_{ij}^{k+1}}\}, \quad k = 1, 2, \ldots,$$

after the kth bounce. A simple geometrical observation yields the following recursive equations for these angles:

$$\theta_k' = \tan^{-1}\left[\frac{\sqrt{\eta^2-1} - \sin(\sum_{i=1}^{k-1}\theta_i')}{1+\cos(\sum_{i=1}^{k-1}\theta_i')}\right], \quad \theta_k = \sum_{i=1}^{k-1}\theta_i + 2\theta_k',$$

where $\eta = \rho_{ij}'/\rho_{ij}$. We note that the relative position vector $p_{i/j}$ may not reach the points Q_{ij}^k or R_{ij}^k, depending on the behavior of the other relative states. However, the new vectors NQ_{ij}^k and NR_{ij}^k, will lead to even larger values for the sequence θ_k by (21)–(23) due to the presence of other relative states (see Fig. 6). In fact, the presence of other relative states will generally reduce the total number of required bounces. Let us clarify further this last observation via an example. Suppose that the position vector $p_{i/j}$ has assumed the vector value Q_{ij}^k and then R_{ij}^k; and suppose that it now is heading toward the final relative position vector $p_{i/j}(N)$. Meanwhile, let another relative state (say $p_{i/j'}$) violate its corresponding constraint set. Observe that the vector $p_{i/j}$ may not assume the value Q_{ij}^{k+1}, and thus the trajectory depicted by the solid line in Fig. 6 (without an arrow) becomes infeasible. Instead, the vector $p_{i/j}$ has to evolve along a different trajectory (solid line with an arrow in Fig. 6) that satisfies (21) and (22). As $p_{i/j'}$ assumes the value $R_{i/j'}$, the vector $p_{i/j}$ arrives at the point P and then is directed toward $p_{i/j}(N)$. Thus, we observe that $p_{i/j}$ will assume the value NQ_{ij}^{k+1}, thereby yielding our previous claim that $N\theta_k \geq \theta_k$. For a fixed value of ρ_{ij}' and when Q_{ij}^{K+1} is $p_{i/j}(N)$ (see Fig. 7), one has

$$\sum_{i=1}^{K-1}\theta_i + \theta_K' \geq \pi/2.$$

Since for an n-spacecraft reconfiguration at most $n(n-1)/2$ relative states constraints are involved, at most $Kn(n-1)/2$ bounces will suffice for the entire reconfiguration. □

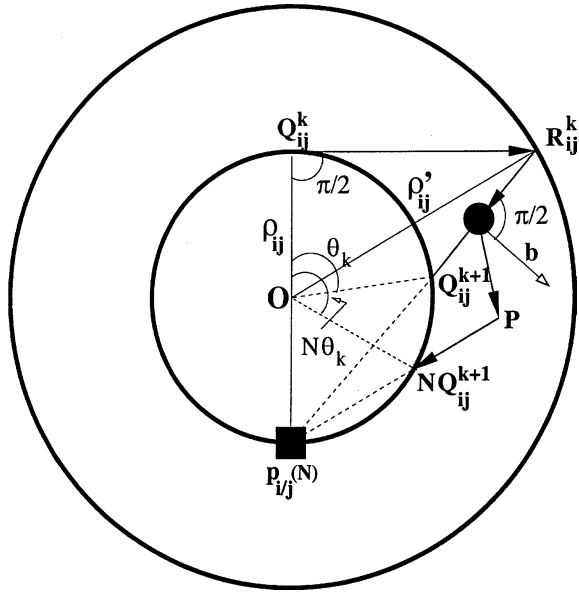

Fig. 6. New values for Q_{ij}^{k+1} and θ_k (NQ_{ij}^{k+1} and $N\theta_k$, respectively) due to the close proximity of another relative position vector.

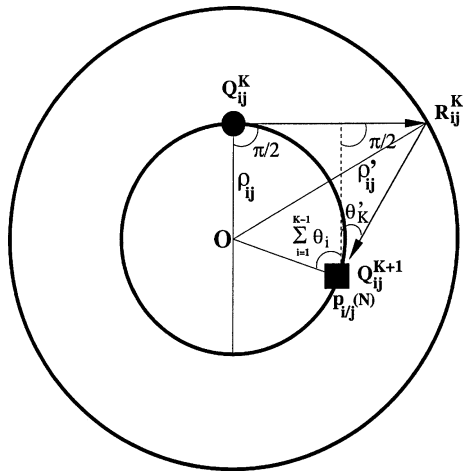

Fig. 7. The case where the inequality $\Sigma_{i=1}^{K-1}\theta_i + \theta'_K \geq \pi/2$ holds.

Table 1. K vs. η.

η	1.0001	1.001	1.01	1.1	1.5	2	10	100
K	218	67	20	6	3	2	2	2

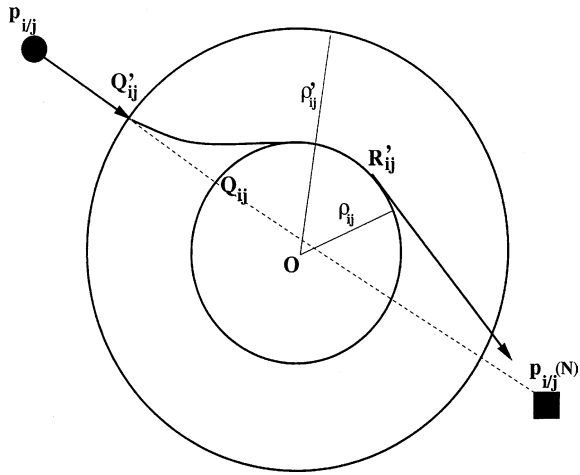

Fig. 8. Variation of the BB Algorithm: The bounced state $p_{i/j}$ follows an smooth path joining Q'_{ij} and R'_{ij}.

Table 1 tabulates the required values of the parameter K for various values of η in Theorem 3.1. Note that each relative state requires at most two bounces during reconfiguration when ρ'_{ij} is sufficiently large ($\rho'_{ij} \geq 1.733\rho_{ij}$). At the same time, when $\eta \approx 1$, the bouncing trajectories resemble the smoother reconfiguration trajectories (see Ref. 15) at the expense of possibly a higher number of required bounces (Fig. 8).

4. Simulation Results

We illustrate now the behavior of the BB Algorithm via an example. The example involves a five-spacecraft reconfiguration that is particularly relevant to the NASA Terrestrial Planet Finder mission (Ref. 1). We assume that each spacecraft has unit mass, that the required minimum distance between any pair of spacecraft ρ_{ij}, for $i, j = 1, \ldots, 5$ and $i \neq j$, is 3 length units, and that the parameters ρ'_{ij}, for $i, j = 1, \ldots, 5$ and $i \neq j$, employed in Phase III of the BB Algorithm are twice the corresponding ρ_{ij}. The initial and final conditions for this example are chosen such that

the first and second spacecraft' and also the third and fourth spacecraft' have to interchange their inertial positions, respectively. The initial and the final states of the fifth spacecraft are chosen such that it is an obstacle for the reconfiguration of the other four spacecraft. Specifically, we have the initial conditions

$$x_1(t_0) = [0, 0, 0, 0, 0, 0]^T, \quad x_2(t_0) = [10, 10, 10, 0, 0, 0]^T,$$
$$x_3(t_0) = [10, 0, 0, 0, 0, 0]^T, \quad x_4(t_0) = [0, 10, 10, 0, 0, 0]^T,$$
$$x_5(t_0) = [0, 0, 10, 0, 0, 0]^T,$$

and the ("desired final conditions")

$$x_1(t_f) - x_2(t_f) = [10, 10, 10, 0, 0, 0]^T,$$
$$x_1(t_f) - x_3(t_f) = [10, 0, 0, 0, 0, 0]^T,$$
$$x_1(t_f) - x_4(t_f) = [0, 10, 10, 0, 0, 0]^T,$$
$$x_1(t_f) - x_5(t_f) = [0, 0, 10, 0, 0, 0]^T,$$

with $t_0 = 0$ and t_f is unspecified. Figure 9 depicts the five spacecraft trajectories (inertial positions) after applying the BB Algorithm. As shown in this figure, each spacecraft moves initially from its initial state I_i, consistent with reaching the desired final relative state $F_i, i = 1, 2, \ldots, 5$. Shortly thereafter, all spacecraft enter their respective collision zones, reaching the points $Q_{1i}, i = 1, 2, \ldots, 5$. In particular, we note that the relative states $x_{1/3}, x_{1/5}, x_{2/4}, x_{4/5}$ are about to violate the corresponding

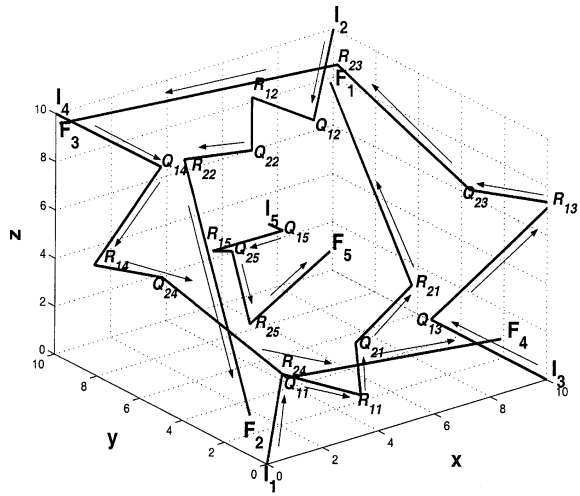

Fig. 9. Example of five-spacecraft reconfiguration trajectories under the BB Algorithm.

constraints of the form (7). Then, the algorithm proceeds to choose one of the violating relative states (say $x_{4/5}$) as $x_{p/q}$ and set the other states as $x_{i/j}$, to solve the quadratic program (18)–(23) for the control forces $u(k)$. This causes the multiple spacecraft to move away from each other, until the fourth and fifth spacecraft position vectors assume the vector values R_{14} and R_{15}, respectively. We note that the relative position vector $p_{4/5}$ satisfies $\|p_{4/5}\| = 6$ at R_{14} and R_{15}. Now, starting from $R_{1i}, i = 1, 2, \ldots, 5$, each spacecraft attempts to move toward its final desired state. Shortly thereafter, the relative state $x_{2/5}$ comes close to the violation of its constraint set at Q_{22} and Q_{25}. As in the previous bounce, the BB Algorithm calculates the required control forces by solving the quadratic program (18)–(23); however, this time the relative state vectors $x_{2/5}$ and $x_{4/5}$ replace the corresponding $x_{p/q}$ and $x_{s/t}$. This leads each spacecraft pair to move away from each other until the second and fifth spacecraft reach the vector values R_{22} and R_{25}, respectively, where $\|p_{2/5}\| = 6$. Now, starting from the points $R_{2i}, i = 1, 2, \ldots, 5$, each spacecraft moves consistently with reaching its terminal states. As illustrated in Fig. 9, only two bounces suffice for the complete reconfiguration. In fact, our extensive simulations for three-to-five spacecraft missions suggest that generally the subroutine STALEMATE of Section 3.4 does not need to be invoked in the execution of the BB Algorithm.

5. Concluding Remarks

Due to the inherent nonconvex nature of the multiple-spacecraft collision-free reconfiguration problem in deep space, in this paper we considered a heuristically motivated approach for its solution. Then, we formalized the resulting approach and proposed an algorithm for the multiple-spacecraft reconfiguration that is built around solving convex quadratic programs. Moreover, we elaborated on the convergence properties of this reconfiguration algorithm for distributed space systems consisting of an arbitrary number of spacecraft. We note that an attractive feature of the proposed algorithm is its relative insensitivity to the sudden appearance or disappearance of a particular set of dynamic and static constraints during the course of the reconfiguration.

References

1. ANONYMOUS, TPF *Book: Origins of Stars, Planets, and Life*, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, 1999.

2. Wang, P. K. C., and Hadaegh, F. Y., *Coordination and Control of Multiple Microspacecraft Moving in Formation*, Journal of the Astronautical Sciences, Vol. 44, pp. 315–355, 1996.
3. Tomlin, C., Pappas, G. J., and Sastry, S., *Conflict Resolution for Air Traffic Management: A Study in Multi-Agent Hybrid Systems*, IEEE Transactions on Automatic Control, Vol. 43, pp. 509–521, 1998.
4. Mesbahi, M., and Hadaegh, F. Y., *Mode and Logic-Based Switching for the Formation Flying Control of Multiple Spacecraft*, Journal of the Astronautical Sciences, Vol. 49, pp. 443–468, 2001.
5. Frazzoli, E., Mao, Z. H., Oh, J. H., and Feron, E., *Aircraft Conflict Resolution via Semidefinite Programming*, Journal of Guidance, Control, and Dynamics, Vol. 24, pp. 79–86, 2001.
6. Kang, W., Sparks, A., and Banda, S., *Coordinated Control of Multi-Satellite Systems*, Journal of Guidance, Control, and Dynamics, Vol. 24, pp. 360–368, 2001.
7. Mesbahi, M., and Hadaegh, F. Y., *Formation Flying Control of Multiple Spacecraft via Graphs, Matrix Inequalities, and Switching*, Journal of Guidance, Control, and Dynamics, Vol. 24, pp. 369–377, 2001.
8. Beard, R., Lawton, J., and Hadaegh, F. Y., *A Coordination Architecture for Spacecraft Formation Control*, IEEE Transactions on Control Systems Technology, Vol. 9, pp 777–790, 2001.
9. Sidi, M., *Spacecraft Dynamics and Control*, Cambridge University Press, New York, NY, 1997.
10. Mehra, R. K., and Davis, R. E., *A Generalized Gradient Method for Optimal Control Problems with Inequality Constraints and Singular Arcs*, IEEE Transactions on Automatic Control, Vol. 17, pp. 69–78, 1972.
11. Bryson, A., and Ho, Y. C., *Applied Optimal Control*, Taylor and Francis, New York, NY, 1981.
12. Stengel, R. F., *Optimal Control and Estimation*, Dover, New York, NY, 1994.
13. Vinter, R., *Optimal Control, Birkhaüser*, Boston, Massachusetts, 2000.
14. Richards, A., Schouwenaars, T., How, J. P., and Feron, E., *Spacecraft Trajectory Planning with Collision and Plume Avoidance Using Mixed-Integer Linear Programming*, Journal of Guidance, Control and Dynamics, Vol. 25, pp. 755–764, 2002.
15. Kim, Y., Mesbahi, M., and Hadaegh, F. Y., *Dual-Spacecraft Formation Flying in Deep Space: Optimal Collision-Free Reconfigurations*, Journal of Guidance, Control, and Dynamics, Vol. 26, pp. 375–379, 2003.
16. Anonymous, *Optimization Toolbox User's Guide*, The MathWorks, Natick, Massachusetts, 2003.